AF$ IFW

Practitioner's Docket No. NAI1P279/01.024.01 ***PATENT***

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Geoff Grindrod et al.

Application No.: 09/853,360 Group No.: 2121
Filed: 05/10/2001 Examiner: Hirl, Joseph P.
For: SYSTEM AND METHOD FOR CUSTOMIZING AND PROCESSING BUSINESS LOGIC RULES IN A BUSINESS PROCESS SYSTEM

Mail Stop Appeal Briefs – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION--37 C.F.R. § 1.192)

1. Transmitted herewith, in triplicate, is the APPEAL BRIEF in this application, with respect to the Notice of Appeal filed on June 17, 2004.

2. STATUS OF APPLICANT

This application is on behalf of other than a small entity.

CERTIFICATION UNDER 37 C.F.R. §§ 1.8(a) and 1.10*

*(When using Express Mail, the Express Mail label number is **mandatory**; Express Mail certification is optional.)*

I hereby certify that, on the date shown below, this correspondence is being:

MAILING

✓ deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

37 C.F.R. § 1.8(a)

✓ with sufficient postage as first class mail.

37 C.F.R. § 1.10*

_ as "Express Mail Post Office to Addressee"
Mailing Label No. _____ **(mandatory)**

TRANSMISSION

_ facsimile transmitted to the Patent and Trademark Office, (703) _____ - _____.

Erica L. Farlow
Signature

Date: 6/18/04

Erica L. Farlow
(type or print name of person certifying)

** Only the date of filing (' 1.6) will be the date used in a patent term adjustment calculation, although the date on any certificate of mailing or transmission under ' 1.8 continues to be taken into account in determining timeliness. See ' 1.703(f). Consider "Express Mail Post Office to Addressee" (' 1.10) or facsimile transmission (' 1.6(d)) for the reply to be accorded the earliest possible filing date for patent term adjustment calculations.*

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 C.F.R. § 1.17(c), the fee for filing the Appeal Brief is:

other than a small entity	$330.00

Appeal Brief fee due $330.00

4. EXTENSION OF TERM

The proceedings herein are for a patent application and the provisions of 37 C.F.R. § 1.136 apply.

Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal brief fee	$330.00
Extension fee (if any)	$0.00

TOTAL FEE DUE $330.00

6. FEE PAYMENT

Attached is a check in the amount of $330.00.

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

If any additional extension and/or fee is required, and if any additional fee for claims is required, charge Deposit Account No. 50-1351 (Order No. NAI1P279).

Signature of Practitioner

Reg. No.: 41,429
Tel. No.: 408-971-2573
Customer No.: 28875

Kevin J. Zilka
Silicon Valley IP Group, PC
P.O. Box 721120
San Jose, CA 95172-1120
USA

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the application of	)
	) Ex.: Hirl, Joseph
Grindrod et al.	)
	) Art Unit: 2121
Application No. 09/853,360	)
	)
Filed: May 10, 2001	)
	)
For: SYSTEM AND METHOD FOR CUSTOMIZING AND PROCESSING BUSINESS LOGIC RULES IN A BUSINESS PROCESS SYSTEM	)

Commissioner for Patents
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals and Interferences

APPELLANT'S BRIEF (37 C.F.R. § 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on June 17, 2004.

The fees required under § 1.17, and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. § 1.192(a))

This brief contains these items under the following headings, and in the order set forth below (37 C.F.R. § 1.192(c)):

I REAL PARTY IN INTEREST

II RELATED APPEALS AND INTERFERENCES

III STATUS OF CLAIMS

IV STATUS OF AMENDMENTS

V SUMMARY OF INVENTION

VI ISSUES

VII GROUPING OF CLAIMS

VIII ARGUMENTS

APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

The final page of this brief bears the practitioner's signature.

I REAL PARTY IN INTEREST (37 C.F.R. § 1.192(c)(1))

The real party in interest in this appeal is Networks Associates Technology, Inc.

II RELATED APPEALS AND INTERFERENCES (37 C.F.R. § 1.192(c)(2))

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no other such appeals or interferences.

III STATUS OF CLAIMS (37 C.F.R. § 1.192(c)(3))

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1, 3-12, 14-23, 25-37, and 39-68.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims withdrawn from consideration but not canceled: None
2. Claims pending: 1, 3-12, 14-23, 25-37, and 39-68
3. Claims allowed: None
4. Claims rejected: 1, 3-12, 14-23, 25-37, and 39-68

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 3-12, 14-23, 25-37, and 39-68

IV STATUS OF AMENDMENTS (37 C.F.R. § 1.192(c)(4))

As to the status of any amendment filed subsequent to final rejection, there are no such amendments after final.

V SUMMARY OF INVENTION (37 C.F.R. § 1.192(c)(5))

A method and system is disclosed for customizing business rules of a business logic application. In use, a content page is served to a client browser of a client by a server, the content page allows for entering and modifying of data relating to a business logic rule. Further, data is generated by the server according to a predefined format from information received via the content page. As set forth in Figure 8 and the related description, the generated data is automatically committed in the predefined format into a database. The database stores data including data relating to business logic rules for implementing business logic as entries in the database. The generated data is committed into a corresponding entry in the database. See database 130 of Figure 1, for example. Upon the committing, the committed database business rule entry is ready for execution by the business logic application. Further, the entering of data includes selectively enabling and disabling the business logic rule via the content page provided by the server.

VI ISSUES (37 C.F.R. § 1.192(c)(6))

Issue #1: The Examiner has rejected Claims 1, 3-12, 14-23, 25-37, and 39-68 under 35 U.S.C. §102(e) as being anticipated by Helgeson et al. (US Pub. 2002/0049749).

VII GROUPING OF CLAIMS (37 C.F.R. § 1.192(c)(7))

The claims of the following groups do not stand or fall together. Following is the grouping of claims. In the following section, appellant explains why the claims of each group are believed to be separately patentable.

Issue # 1: Grouping of Claims

Group #1: Claims 1, 4-6, 11-12, 15-17, 22-23, 26-28, 33-37, 40-41, 45-49, 52, 54-61, and 63-68

Group #2: Claims 3, 14, 25, and 39

Group #3: Claims 7, 18, and 29

Group #4: Claims 8-10, 19-21, 30-32, and 42-44

Group #5: Claims 50 and 62

Group #6: Claims 51 and 53

VIII ARGUMENTS (37 C.F.R. § 1.192(c)(8))

Issue #1:

The Examiner has rejected Claims 1, 3-12, 14-23, 25-37, and 39-68 under 35 U.S.C. §102(e) as being anticipated by Helgeson et al. (US Pub. 2002/0049749).

Group #1: Claims 1, 4-6, 11-12, 15-17, 22-23, 26-28, 33-37, 40-41, 45-49, 52, 54-61, and 63-68

In the Examiner's most recent response, it is mentioned that the Examiner is obligated to "interpret each claim in the broadest reasonable sense." The Examiner then relies on this statement to support the assertion that "selectively enabling and disabling business logic rules means implementing the high level object oriented technology which is binary logic based, meaning that is it selectively enabled and disabled (on, off; 1, 0; etc.)"

This assertion, however, is not "reasonable," as the excerpt cited by the Examiner merely suggests that the business application of Helgeson may be implemented using objected oriented technology. The Examiner continues by arguing that the use of objected oriented technology implies use of 1's and 0's (like any digital framework). However, if Helgeson implicitly discloses 1's and 0's, it would require an unreasonable leap to conclude that business logic rules

are selectively enabled and disabled, especially since the Examiner is relying on an allegedly implicit teaching that the disclosed business application includes business logic rules.

Specifically, the nature of the Examiner's arguments is particularly unreasonable in view of the specific context in which the foregoing limitations are claimed. Specifically, appellant teaches and claims "serving a content page to a client browser of a client by a server, the content page allowing for entering and modifying of data relating to a business logic rule," "wherein the entering of data includes selectively enabling and disabling the business logic rule via the content page provided by the server."

Thus, following the Examiner's logic, Helgeson would have to enter the 1's and 0's via a content page, as claimed. Of course, however, 1's and 0's are an invisible digital framework on which computers are based, and can simply not be entered via a content page, as required by appellant's claims. In view of this impossibility and obvious deficiencies, the Examiner's interpretation of Helgeson and application thereof to appellant's claims is not "reasonable."

In summary, Helgeson fails to disclose, teach or suggest any sort of content page that allows entry of data with the specific capability of selectively enabling and disabling the business logic rule, in the specific context of "serving a content page to a client browser of a client by a server, the content page allowing for entering and modifying of data relating to a business logic rule." Instead, Helgeson teaches selection of a command with which processing (not "enabling and disabling") of a web document (not "a business logic rule") is carried out.

Only appellant teaches and claims such a unique ability of enabling and disabling the business logic rule in the specific context of the remaining claim limitations.

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, the identical invention must be shown in as complete detail as contained in the claim. *Richardson v.*

*Suzuki Motor Co.*868 F.2d 1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

This criterion has simply not been met by the Helgeson reference, for the reasons set forth hereinabove.

Group #2: Claims 3, 14, 25, and 39

With respect to the present grouping, the Examiner relies on the following excerpt to make a prior art showing of appellant's claimed "wherein data for each business rule includes general information data, condition data, action data, and schedule data." See Claim 3 et al.

> "[0019] A second Internet-based method is disclosed for implementing a business application using high-level object oriented technology and frameworks, the method providing a client input device having a user interface (UI) wherein the user selects a command and a display device whereby results are displayed, and transmits the command to a server computer hosting a business application management system platform, which includes a WDK Web interface server for receiving the user selected command and for processing a web document that is a custom presentation of information. This second Internet-based method also includes receiving at the client input device a display of results relating to the command, the results obtained by an information distributor server electronically coupled to the WDK Web interface server for generating metadata for a business object, for storing the metadata in a metadata database, for querying the metadata database when asked to do so by a requestor, and for providing the results of a match to a query to the requestor; whereby the business

application is available via the internet to assist a user in performing a specific business operation which requires location of and use of business objects and display of results of the specific business operation to the user in a dynamically selectable format."

However, such excerpt fails to disclose, teach or even suggest any sort of "business rule includ[ing] general information data, condition data, action data, and schedule data" (emphasis added), as claimed.

Group #3: Claims 7, 18, and 29

With respect to the present group, the Examiner relies on the foregoing excerpt to make a prior art showing of appellant's claimed "wherein said serving the content page to the client browser includes serving an expression builder content page for entering and modifying of the data relating to the business logic rule expressed as an expression having symbols to be resolved when an instance of the business rule is one of created and executed by the business logic application." See Claim 7 et al. Further, the Examiner argues that "expression builder is a grouping of software, all software operates with symbols."

However, the foregoing excerpt and the Examiner's comment fail to address appellant's claimed "wherein said serving the content page to the client browser includes serving an expression builder content page for entering and modifying of the data relating to the business logic rule expressed as an expression having symbols to be resolved when an instance of the business rule is one of created and executed by the business logic application." Emphasis added. The Examiner is clearly not taking into consideration all of appellant's claim limitations.

Group #4: Claims 8-10, 19-21, 30-32, and 42-44

Regarding Group #4, the Examiner relies on the foregoing excerpt to make a prior art showing of appellant's claimed data that selectively includes a condition, action, and schedule "expressed as

an expression via the expression builder content page." See Claims 8-10 et al. Further, the Examiner argues that "all data is in one form or another action data albeit potential and such information will be operated on (expressed) by software."

However, the foregoing excerpt and the Examiner's comment fail to address that appellant's claims require that the schedule, action, and condition data includes data "expressed as an expression via the expression builder content page" (emphasis added).

It appears that the Examiner is not taking into consideration the full weight of all of appellant's claim limitations.

Group #5: Claims 50 and 62

Regarding Group #5, the Examiner relies on the foregoing excerpt to make a prior art showing of appellant's claimed "resolving embedded pre-queue symbols in expressions of the event job corresponding to the business rule." See Claims 50 et al. Further, the Examiner argues that "all software has embedded symbols of whatever vintage."

However, the foregoing excerpt and the Examiner's comment fail to address that appellant's claims require "resolving embedded pre-queue symbols in expressions of the event job corresponding to the business rule" (emphasis added). The only mention in Helgeson of any sort of queue is in the following paragraph.

> "[0862] Key to the design of the Interconnect is the notion of pluggable transport protocols. To accommodate this, the Delivery Service has 2 components (1) Delivery Service (2) Persistent Message Manager. The Delivery Service writes messages to outbound queues (if the message needs to be delivered to an external system), the Persistent Message Manager polls out bound queues to deliver the message to the host the message is intended for. The persistent

Message Manager has the uses pluggable transport protocol. For implementing a protocol using RMI a class needs to be written implementing IPMTransport. The Persistent Message Manager (PMM) acts as the listener for receiving messages. Messages received are put into inbound queues, the Delivery Service delivers messages from the inbound queues to the Subscribers."

Such excerpt, however, fails to meet the claim limitations emphasized hereinabove. The necessary anticipation criteria has simply not been met.

Group #6: Claims 51 and 53

Regarding Group #6, the Examiner relies on FIG. 2 of Helgeson to make a prior art showing of appellant's claimed "wherein said resolving embedded pre-queue symbols in expressions of the event job comprises: parsing the expression hierarchically; and resolving the symbols in a recursive manner." See Claims 51 et al. Further, the Examiner argues that "to one of ordinary skill in the art, parsing is a genetic skill in computer science."

However, the foregoing excerpt and the Examiner's comment fail to address that appellant's claims require "wherein said resolving embedded pre-queue symbols in expressions of the event job comprises: parsing the expression hierarchically; and resolving the symbols in a recursive manner" (emphasis added). The only mention in Helgeson of any sort of queue is in the paragraph cited above. Such excerpt, however, fails to meet the foregoing claim limitations emphasized hereinabove.

In view of the remarks set forth hereinabove, all of the independent claims are deemed allowable, along with any claims depending therefrom.

IX APPENDIX OF CLAIMS (37 C.F.R. § 1.192(c)(9))

The text of the claims involved in the appeal is:

1. (Previously Amended) A method for customizing business rules of a business logic application, comprising:

serving a content page to a client browser of a client by a server, the content page allowing for entering and modifying of data relating to a business logic rule;

generating data by the server according to a predefined format from information received via the content page; and

automatically committing the generated data in the predefined format into a database, the database storing data including data relating to business logic rules for implementing business logic as entries in the database, the generated data being committed into a corresponding entry in the database;

wherein upon said committing, the committed database business rule entry is ready for execution by the business logic application;

wherein the entering of data includes selectively enabling and disabling the business logic rule via the content page provided by the server.

2. (Cancelled)

3. (Original) The method for customizing business rules of a business logic application of claim 1, wherein data for each business rule includes general information data, condition data, action data, and schedule data.

4. (Original) The method for customizing business rules of a business logic application of claim 1, further comprising, prior to said committing, automatically verifying the entry corresponding to the business logic rule in the predefined format using DTDs (Document Type Definitions).

5. (Original) The method for customizing business rules of a business logic application of claim 1, wherein said committing the entered data according to the predefined format includes committing the entered data according to a predefined XML format.

6. (Original) The method for customizing business rules of a business logic application of claim 5, further comprising, prior to said committing, automatically verifying the entry corresponding to the business logic rule in the predefined XML format using DTDs (Document Type Definitions).

7. (Original) The method for customizing business rules of a business logic application of claim 1, wherein said serving the content page to the client browser includes serving an expression builder content page for entering and modifying of the data relating to the business logic rule expressed as an expression having symbols to be resolved when an instance of the business rule is one of created and executed by the business logic application.

8. (Original) The method for customizing business rules of a business logic application of claim 7, wherein data for each business rule includes schedule data, the schedule data selectively includes schedule expressed as an expression via the expression builder content page.

9. (Original) The method for customizing business rules of a business logic application of claim 7, wherein data for each business rule includes action data, the action data selectively includes action data expressed as an expression via the expression builder content page.

10. (Original) The method for customizing business rules of a business logic application of claim 7, wherein data for each business rule includes condition data, the condition data selectively includes a condition expressed as an expression via the expression builder content page.

11. (Original) The method for customizing business rules of a business logic application of claim 1, wherein said serving the content page to the client browser includes serving a business rules management content page for displaying business rules stored as entries in the

database and for allowing one of selecting to create a new business rule and selecting to modify an existing business rule.

12. (Previously Amended) A computer program product for customizing business rules of a business logic application, comprising:

computer code that serves a content page to a client browser of a client, the content page allowing for entering and modifying of data relating to a business logic rule;

computer code that generates data according to a predefined format from information received via the content page;

computer code that automatically commits the generated data in the predefined format into a database that stores data including data relating to business logic rules for implementing business logic as entries in the database, the computer code commits the generated data into a corresponding entry in the database such that the committed database business rule entry is ready for execution by the business logic application; and

a computer readable medium that stores said computer codes;

wherein computer code that serves the content page also allows selectively enabling and disabling the business logic rule via the content page.

13. (Cancelled)

14. (Original) The computer program product for customizing business rules of a business logic application of claim 12, wherein data for each business rule includes general information data, condition data, action data, and schedule data.

15. (Original) The computer program product for customizing business rules of a business logic application of claim 12, further comprising computer code that automatically verifies the entry corresponding to the business logic rule in the predefined format using DTDs (Document Type Definitions).

16. (Original) The computer program product for customizing business rules of a business logic application of claim 12, wherein computer code that commits the entered data

according to the predefined format includes computer code that commits the entered data according to a predefined XML format.

17. (Original) The computer program product for customizing business rules of a business logic application of claim 16, further comprising computer code that automatically verifies the entry corresponding to the business logic rule in the predefined XML format using DTDs (Document Type Definitions).

18. (Original) The computer program product for customizing business rules of a business logic application of claim 12, wherein the computer code that serves the content page includes computer code that serves an expression builder content page for entering and modifying of the data relating to the business logic rule expressed as an expression having symbols to be resolved when an instance of the business rule is one of created and executed by the business logic application.

19. (Original) The computer program product for customizing business rules of a business logic application of claim 18, wherein data for each business rule includes schedule data, the schedule data selectively includes schedule expressed as an expression via the expression builder content page.

20. (Original) The computer program product for customizing business rules of a business logic application of claim 18, wherein data for each business rule includes action data, the action data selectively includes action data expressed as an expression via the expression builder content page.

21. (Original) The computer program product for customizing business rules of a business logic application of claim 18, wherein data for each business rule includes condition data, the condition data selectively includes a condition expressed as an expression via the expression builder content page.

22. (Original) The computer program product for customizing business rules of a

business logic application of claim 12, wherein the computer code that serves the content page to the client browser includes computer code that serves a business rules management content page for displaying business rules stored as entries in the database and for allowing one of selecting to create a new business rule and selecting to modify an existing business rule.

23. (Previously Amended) A business logic application system adapted for customizing business rules, comprising:

a client having a client browser;

a database for storing data including data relating to business logic rules for implementing business logic as entries in the database; and

a server having a web server adapted to serve at least one content page to the client browser for entering and modifying of the data of a business logic rule corresponding to an entry in the database,

wherein said server is adapted to automatically commit an entry corresponding to the business logic rule into the database according to a predefined format after modifications via the content page at said client browser and wherein the committed database business rule entry is ready for execution upon commitment by said server;

wherein each business logic rule can be selectively enabled via the content page provided by the server at the client browser.

24. (Cancelled)

25. (Original) The business logic application system of claim 23, wherein data for each business rule includes general information data, condition data, action data, and schedule data.

26. (Original) The business logic application system of claim 23, wherein said server is adapted to verify the entry corresponding to the business logic rule in the predefined format using DTDs (Document Type Definitions).

27. (Original) The business logic application system of claim 23, wherein the predefined format for committing the entry corresponding to the business logic rule into the database is a

predefined XML format.

28. (Original) The business logic application system of claim 27, wherein said server is adapted to verify the entry corresponding to the business logic rule in the predefined XML format using DTDs (Document Type Definitions).

29. (Original) The business logic application system of claim 23, wherein said server is further adapted to serve an expression builder content page to the client browser for entering and modifying of the data of the business logic rule expressed as an expression having symbols to he resolved when an instance of the business rule is one of created and executed.

30. (Original) The business logic application system of claim 29, wherein data for each business rule includes schedule data, the schedule data selectively includes a schedule expressed as an expression via the expression builder content page.

31. (Original) The business logic application system of claim 29, wherein data for each business rule includes action data, the action data selectively includes action data expressed as an expression via the expression builder content page.

32. (Original) The business logic application system of claim 29, wherein data for each business rule includes condition data, the condition data selectively includes a condition expressed as an expression via the expression builder content page.

33. (Original) The business logic application system of claim 23, wherein said server is further adapted to serve a business rules management content page to the client browser for displaying business rules stored as entries in the database and for allowing one of selecting to create a new business rule and selecting to modify an existing business rule.

34. (Previously Amended) An application/web server for implementing a business logic application system adapted for customizing business rules, comprising:

a web server in communication with a client browser of a client and adapted to serve

at least one non-programmatic interactive user page to the client browser for obtaining data for a customized business logic rule; and

an application server in communication with a database containing data relating to business logic rules for implementing business logic as entries in the database, wherein the data obtained for the customized business logic rule corresponds to an entry in the database,

wherein the application server is adapted to dynamically and automatically commit the data as an entry corresponding to the customized business logic rule into the database in a predefined format after obtaining the data via the user page at said client browser and wherein the committed database business rule entry is ready for execution upon commitment by said application server;

wherein each business logic rule in the database can be selectively enabled via the at least one non-programmatic interactive user page.

35. (Original) The application/web server for implementing a business logic application system of claim 34, wherein said application server is adapted to verify the entry corresponding to the customized business logic rule in the predefined format using DTDs (Document Type Definitions).

36. (Original) The application/web server for implementing a business logic application system of claim 34, wherein the predefined format for committing the entry corresponding to the business logic rule into the database is a predefined XML format.

37. (Original) The application/web server for implementing a business logic application system of claim 36, wherein said application server is adapted to verify the entry corresponding to the business logic rule in the predefined XML format using DTDs (Document Type Definitions).

38. (Cancelled)

39. (Original) The application/web server for implementing a business logic application system of claim 34, wherein data for each business rule includes general information data,

condition data, action data, and schedule data.

40. (Original) The application/web server for implementing a business logic application system of claim 34, wherein the at least one non-programmatic interactive user page includes a business rules management page for displaying business rules stored as entries in the database and for allowing one of selecting to create a new business rule and selecting to modify an existing business rule.

41. (Original) The application/web server for implementing a business logic application system of claim 34, wherein the at least one non-programmatic interactive user page includes an expression builder page expressing data of the business logic rule as an expression having at least one symbol that is to be resolved when an instance of the business rule is one of created and executed.

42. (Original) The application/web server for implementing a business logic application system of claim 41, wherein data for each business rule includes schedule data, the schedule data selectively includes a schedule expressed as an expression via the expression builder page.

43. (Original) The application/web server for implementing a business logic application system of claim 41, wherein data for each business rule includes action data, the action data selectively includes action data expressed as an expression via the expression builder page.

44. (Original) The application/web server for implementing a business logic application system of claim 41, wherein data for each business rule includes condition data, the condition data selectively includes a condition expressed as an expression via the expression builder page.

45. (Previously Amended) A method for processing business logic rules of a business process system. comprising:

writing an event job into a job queue for each occurrence of an event having at least one business rule based on occurrence thereof;

creating a business rule instance for each business rule corresponding to the event

job;

testing conditions of each business instance;

writing the business rule instance into the job queue corresponding to each business instance for which the conditions testing succeeds;

deleting the event job from the job queue;

executing the business rule instance; and

deleting the business rule instance from the job queue;

wherein the business logic rules can be selectively enabled via a content page provided by a server at a client browser.

46. (Original) The method for processing business logic rules of claim 45, further comprising scheduling the business rule instance, wherein said executing the business rule instance is according to said scheduling.

47. (Original) The method for processing business logic rules of claim 46, wherein said scheduling the business rule instance is selected from the group consisting of delaying job execution, rescheduling job execution, scheduling repeat executions, and suspending execution.

48. (Original) The method for processing business logic rules of claim 45, wherein said executing the business rule instance comprises:

testing conditions of the business rule instance;

if the instance conditions testing fails, deleting the business rule instance from the job queue; and

if the instance conditions testing succeeds:

executing actions specified by the business rule instance, and

deleting the business rule instance from the job queue.

49. (Original) The method for processing business logic rules of claim 48, wherein said executing actions specified by the business rule instance comprises:

scheduling execution using an execution schedule as determined according to scheduling data of the business rule instance; and

executing actions specified by the business rule instance according to the execution schedule.

50. (Original) The method for processing business logic rules of claim 45, further comprising resolving embedded pre-queue symbols in expressions of the event job corresponding to the business rule.

51. (Original) The method for processing business logic rules of claim 50, wherein said resolving embedded pre-queue symbols in expressions of the event job comprises:

parsing the expression hierarchically; and

resolving the symbols in a recursive manner.

52. (Original) The method for processing business logic rules of claim 45, further comprising resolving embedded post-queue symbols in expressions of the business rule instance.

53. (Original) The method for processing business logic rules of claim 52, wherein said resolving embedded post-queue symbols in expressions of the business rule instance comprises:

parsing the expression hierarchically; and

resolving the symbols in a recursive manner.

54. (Original) The method for processing business logic rules of claim 45, further comprising:

monitoring for incoming notification events; and

reporting the incoming notification events to the business process system.

55. (Original) The method for processing business logic rules of claim 45, wherein said executing the business rule instance includes selectively transmitting an outgoing notification.

56. (Original) The method for processing business logic rules of claim 55, wherein the outgoing notification is selected from the group consisting of mail, pager notification, Telalert, and NT network message notifications.

57. (Original) A business logic application system for implementing business logic rules, comprising:

a job queue module for maintaining a job queue and processing jobs in the job queue; and

a notification module in communication with said job queue module for monitoring for incoming notification events and reporting the incoming notification events to the job queue module,

wherein said job queue module writes an event job corresponding to a business logic rule to the job queue upon receiving an incoming notification event that matches a triggering event of the business logic rule,

said job queue module tests conditions of the business logic rule corresponding to the event job, deletes the event job from the job queue, and, if conditions of the business logic rule are met, writes a business rule instance to the job queue, and

said job queue module processes the business rule instance and deletes the business rule instance from the job queue;

wherein the business logic rules can be selectively enabled via a content page provided by a server at a client browser.

58. (Original) The business logic application system of claim 57, wherein the job queue module processes the business rule instance by scheduling the business rule instance and executing the business rule instance is according to the scheduling.

59. (Original) The business logic application system of claim 58, wherein the scheduling by the job queue module is selected from the group consisting of delaying job execution, rescheduling job execution, scheduling repeat executions, and suspending execution.

60. (Original) The business logic application system of claim 57, wherein the job queue module processes the business rule instance includes testing conditions of the business rule instance, executing actions specified by the business rule instance if the conditions testing succeeds, and deleting the business rule instance form the job queue.

61. (Original) The business logic application system of claim 60, wherein the job queue module executes actions specified by the business rule instance includes scheduling execution according to scheduling data of the business rule instance and executing actions specified by the business rule instance according to the execution scheduling.

62. (Original) The business logic application system of claim 57. wherein the job queue module resolves any embedded pre-queue symbols in expressions of the event job corresponding to the business rule.

63. (Original) The business logic application system of claim 62, wherein the job queue module resolves embedded pre-queue symbols by parsing the expression hierarchically and resolving the symbols in a recursive manner.

64. (Original) The business logic application system of claim 57, wherein the job queue resolves, any embedded post-queue symbols in expressions of the business rule instance.

65. (Original) The business logic application system of claim 64, wherein the job queue module resolves embedded post-queue symbols by parsing the expression hierarchically and resolving the symbols in a recursive manner.

66. (Original) The business logic application system of claim 57, wherein the notification module monitors for incoming notification events and reports the incoming notification events to the job queue module.

67. (Original) The business logic application system of claim 57, wherein the job queue module processes the business rule instance by selectively transmitting an outgoing notification to the notification module.

68. (Original) The business logic application system of claim 67, wherein the notification module processes an outgoing notification task received from the job queue module

by sending out a notification selected from the group consisting of mail, pager notification, Telalert and NT network message notifications.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 971-2573. For payment of any additional fees due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NAI1P279/01.024.01).

Respectfully submitted,

By: _____ Date: 2/18/04

Kevin J. Zilka

Reg. No. 41,429

Silicon Valley IP Group, P.C.
P.O. Box 721120
San Jose, California 95172-1120
Telephone: (408) 971-2573
Facsimile: (408) 971-4660